



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/767,340	01/30/2004	Bryan R. Goring	T8467901US	5223
92030 7590 06/03/2010 Gowling Lafleur Henderson LLP Suite 1600 1 First Canadian Place 100 King Street West Toronto, ON M5X1G5 CANADA				
EXAMINER				
DAO, THUY CHAN				
ART UNIT		PAPER NUMBER		
2192				
MAIL DATE		DELIVERY MODE		
06/03/2010		PAPER		

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

### Office Action Summary

**Application No.**

10/767,340

**Applicant(s)**

GORING ET AL.

**Examiner**

Thuy Dao

**Art Unit**

2192

**Period for Reply** -- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 14 April 2010.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 1-8, 10-28 and 30-42 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-8, 10-28 and 30-42 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 30 January 2004 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
- Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
- Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO/SB08)
- Paper No(s)/Mail Date \_\_\_\_\_
- 4) ☐ Interview Summary (PTO-413)
- Paper No(s)/Mail Date \_\_\_\_\_
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: \_\_\_\_\_

### DETAILED ACTION

1. This action is responsive to the amendment filed on April 14, 2010.
2. Claims 1-8, 10-28, and 30-42 have been examined.

### Response to Arguments

3. New ground of rejection in the previous Office action mailed January 15, 2010 (Remarks, pages 9-13).

As an initial matter, examiner notes that a new ground of rejection was clearly/properly set forth in the Office action mailed January 15, 2010 (page 4, Jensen discloses "the provisioning instructions being customized...") when compared with the Office action mailed July 7, 2009 (page 4, Jensen does not disclose "the provisioning instructions being customized).

Jensen discloses *the provisioning instructions being customized* (e.g., [0032] and [0035], for a particular device/user profile and an identified price/service plan → specific (customized) provisioning instructions are executed).

Jensen does not explicitly disclose *[the provisioning instructions being customized]* for different subsets of versions of the application.

However, in an analogous art, Kjellbert further discloses *[the provisioning instructions being customized]* for different subsets of versions of the application (e.g., [0024]-[0026]) as follows:

"[0025] With reference now to FIG. 1 of the drawings, there is illustrated a provisioning server 200 capable of provisioning objects and applications to client devices 100 in real-time. As noted above, provisioning is the capability to receive a request for an application or object, find a suitable version of the requested application or object and provide the application or object to the requester. The ability to find a suitable version of the requested application or object

accounts for the different formats utilized by the many different types of client devices 100, each with its own characteristics, limitations and configuration. For example, the client devices 100 may include PDAs 100a, workstations and desktop computers 100b, mobile phones 100c and laptops 100d. The characteristics and configurations of each of the different types of client devices 100 are stored in a device profiles database 230 within the provisioning server 200." (requested application has different versions utilized by different types of client devices 100, emphasis added);

"[0026] An application configuration interface 280 serves as a single-point entry into the provisioning server 200 for application providers 420. The application configuration interface 280 allows application providers 420 to configure new services (objects or applications), device type profiles and billing rules specific to the application that is being published through the provisioning server 200. Application descriptors associated with the configured objects or applications, along with the device type profiles and billing rules, are stored in a descriptor database 250. The applications or objects themselves are stored in various application servers 400 accessible to the provisioning server 200." (provisioning server 200 has specific/customized instructions to provision different versions of the requested application from different types of client devices 100, emphasis added).

4. The new ground of rejection based on Krantz (Remarks, pages 13-14).

Jensen discloses *executing by the runtime environment the provisioning instructions for employing the API set* (e.g., FIG. 2, provisioning digital

services/applications according to target device/user profiles and/or price/service plans, [0014]-[0015], [0043]-[0047], [0050]-[0053]).

Neither Jensen nor Kjellberg explicitly discloses *[executing by the runtime environment the provisioning instructions for employing the API set,] by a script interpreter.*

However, in an analogous art, Krantz further discloses *[executing by the runtime environment the provisioning instructions for employing the API set,] by a script interpreter* (an XML/script interpreter/parser/engine for interpreting XML-format files/XML provisioning files) as follows:

[0047] Network provisioning services are potentially available via a variety of media (e.g., WWAN, dial-up, DSL, etc.). A user, through selection rule type "3" recited above (network provider service preference order), potentially specifies a rule that defers selection of a particular media to provisioning service-supplied rules. In an illustrative embodiment, the provisioning service 314 specifies such rules in the form of XML files.

[0063] The following three methods are supported by the common RPC API 303 of the rules engine for the provisioning services 214. A CreateNetworkConfiguration method 434 is similar to the above-described SetNetworkAuthData method 432; however, a network configuration is supplied in XML format via an input parameter. An UpdateNetworkConfiguration method 436 is similar to the CreateNetworkConfiguration method 434 except that a pre-existing configuration is assumed to exist for the identified network. A CreateUserAuthData method 438 is similar to the SetUserAuthData method 428 except the user data is supplied in a passed parameter in XML format.

[0089] In yet another scenario, provided in the fourth row of the table, a rule applied by the rules engine 300 specifies a preference order comprising logical networks. A specific scenario example includes specifying: a corporate network over WISP A over WISP B. A network provider decides which physical network (WLAN over WWAN) to use based upon XML provisioning. Parameters used in such a rule scenario include: XML provisioning files from a wireless provider, business logic to facilitate dynamic network and interface selection. (emphasis added).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine Krantz's teaching into Jensen and Kiellberg's teaching. One would have been motivated to do so to provide network provisioning services by using XML rules files, configuration files, and provisioning files as suggested by Krantz (e.g., [0047], [0063], and [0089]).

### **Claim Rejections – 35 USC §101**

5. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

6. Claim 42 is directed to a computer program product comprising a computer readable medium, which may include transmission medium for provisioning wireless devices 101 in an over-the-air environment (specification, pages 5 and 8).

A computer readable medium product is a tangible physical article or object, some form of matter, which a signal is not. That the other two product classes, machine and composition of matter, require physical matter is evidence that a manufacture was also intended to require physical matter. A signal, a form of energy, does not fall within

either of the two definitions of manufacture. Thus, a signal does not fall within one of the four statutory classes of Sec. 101 – see MPEP 2106

Under the principles of compact prosecution, claim 42 has been examined as the Examiner anticipates the claims will be amended to obviate these 35 USC § 101 issues. For example, - the computer program product comprising: a non-transitory computer readable medium; ... on the non-transitory computer readable medium for obtaining the content; - .

### Double Patenting

7. A rejection based on double patenting of the "same invention" type finds its support in the language of 35 U.S.C. 101 which states that "whoever invents or discovers any new and useful process ... may obtain a patent therefor ..." (Emphasis added). Thus, the term "same invention," in this context, means an invention drawn to identical subject matter. See *Miller v. Eagle Mfg. Co.*, 151 U.S. 186 (1894); *In re Ockert*, 245 F.2d 467, 114 USPQ 330 (CCPA 1957); and *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970).

A statutory type (35 U.S.C. 101) double patenting rejection can be overcome by canceling or amending the conflicting claims so they are no longer coextensive in scope. The filing of a terminal disclaimer cannot overcome a double patenting rejection based upon 35 U.S.C. 101.

The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. A nonstatutory obviousness-type double patenting rejection is appropriate where the conflicting claims are not identical, but at least one examined application claim is not patentably distinct from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s). See, e.g., *In re Berg*, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir.

1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) or 1.321(d) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent either is shown to be commonly owned with this application, or claims an invention made as a result of activities undertaken within the scope of a joint research agreement.

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

Claims 1, 21, 41, and 42 are rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claims 1-3, 8, 11, 15, and 29 of U.S. Patent No. 7,509,658.

### **Claim Rejections – 35 USC §103**

8. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

9. Claims 1-8, 10-28, and 30-42 are rejected under 35 U.S.C. 103(a) as being unpatentable over Jensen (US Patent No. 2004/0261086 A1 in view of Kjellberg (US Patent No. 2003/0084165 A1) and Krantz (US Patent No. 2005/0091357 A1).

#### **Claim 1:**



Jensen discloses *a method for providing customized provisioning of an application on a runtime environment of a terminal* (e.g., FIG. 2, provisioning digital services/applications and deploying/installing the provisioned digital services/applications to target devices 202a-c, [0014]-[0015], [0024]-[0030]),

*the application including content* (e.g., Provisioning Application 208, Database 220, and digital services/applications include a plurality of contents, [0025], [0033]-[0035]),

*having at least one specified content type* (e.g., specified content type such as target devices/user profiles, price/service plans, [0027]-[0033], [0037]-[0042]), *the method comprising the steps of:*

*for each content type, obtaining the content by the runtime environment* (e.g., FIG. 2, retrieving/obtaining target devices/user profiles and price/service plans by runtime environment of Provisioning Server 204, [0024]-[0027], [0031]-[0033]);

*obtaining by the runtime environment a set of provisioning instructions related to the content type* (e.g., [0027][0033], for each device/user profile, price/service plan → executing/obtaining a set or provisioning instructions associated with said device/user profile, price/service plan, [0037]-[0042]),

*the provisioning instructions being customized* (e.g., [0032] and [0035], for particular device/user profile and identified price/service plan → specific provisioning instructions are executed, i.e., customized)

*by distributed provisioning control through the provisioning instructions* (e.g., FIG. 2, execution/control of provisioning instructions and/or provisioning APIs have been distributed over Provisioning Application 208, Provisioning API 222, Adapter 206, but not been hardcoded in target devices 202),

*the provisioning instructions coupled to the application for specifying a provisioning Application Program Interface (API) set for provisioning the content on the terminal* (e.g., FIG. 3, Provisioning Server 204 specifying either Discovery, Subscription, or Delivery Provisioning API set, [0028]-[0031], [0036]-[0039], [0041]-[0043]); *and*

*executing by the runtime environment the provisioning instructions for employing the API set to provision the application according to the specified content*

type (e.g., FIG. 2, provisioning digital services/applications according to target device/user profiles and/or price/service plans, [0014]-[0015], [0043]-[0047], [0050]-[0053]).

Jensen does not explicitly disclose *[the provisioning instructions being customized]* for different subsets of versions of the application.

However, in an analogous art, Kjellbert further discloses *[the provisioning instructions being customized]* for different subsets of versions of the application (e.g., [0024]-[0026]).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine Kjellberg's teaching into Jensen's teaching. One would have been motivated to do so to provision a suitable/new version to client devices in real-time as suggested by Kjellberg (e.g., [0025]-[0026]).

Neither Jensen nor Kjellberg explicitly discloses *[executing by the runtime environment the provisioning instructions for employing the API set,] by a script interpreter*.

However, in an analogous art, Krantz further discloses *[executing by the runtime environment the provisioning instructions for employing the API set,] by a script interpreter* (interpreting XML-format files/XML provisioning files) as follows:

[0047] Network provisioning services are potentially available via a variety of media (e.g., WWAN, dial-up, DSL, etc.). A user, through selection rule type "3" recited above (network provider service preference order), potentially specifies a rule that defers selection of a particular media to provisioning service-supplied rules. In an illustrative embodiment, the provisioning service 314 specifies such rules in the form of XML files.

[0063] The following three methods are supported by the common RPC API 303 of the rules engine for the

provisioning services 214. A CreateNetworkConfiguration method 434 is similar to the above-described SetNetworkAuthData method 432; however, a network configuration is supplied in XML format via an input parameter. An UpdateNetworkConfiguration method 436 is similar to the CreateNetworkConfiguration method 434 except that a pre-existing configuration is assumed to exist for the identified network. A CreateUserAuthData method 438 is similar to the SetUserAuthData method 428 except the user data is supplied in a passed parameter in XML format.

[0089] In yet another scenario, provided in the fourth row of the table, a rule applied by the rules engine 300 specifies a preference order comprising logical networks. A specific scenario example includes specifying: a corporate network over WISP A over WISP B. A network provider decides which physical network (WLAN over WWAN) to use based upon XML provisioning. Parameters used in such a rule scenario include: XML provisioning files from a wireless provider, business logic to facilitate dynamic network and interface selection. (emphasis added).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine Krantz's teaching into Jensen and Kiellberg's teaching. One would have been motivated to do so to provide network provisioning services by using XML rules files, configuration files, and provisioning files as suggested by Krantz (e.g., [0047], [0063], and [0089]).

**Claim 2:**

Jensen discloses *the method according to claim 1, wherein provisioning control of the content is shared between the runtime environment and the application through the coupled provisioning instructions* (e.g., [0025], [0033]-[0035]).

**Claim 3:**

Jensen discloses *the method according to claim 2 further comprising the step of employing a provisioning service to direct the provisioning API, the service configured for recognizing the provisioning instructions* (e.g., [0014]-[0015], [0024]-[0030]).

**Claim 4:**

Jensen discloses *the method according to claim 3 further comprising the step of the service customizing the provisioning process and the associated provisioning API set according to the provisioning instructions* (e.g., [0024]-[0027], [0031]-[0033]).

**Claim 5:**

Jensen discloses *the method according to claim 4, wherein the service is shared by a plurality of the applications* (e.g., [0028]-[0031], [0036]-[0039], [0041]-[0043]).

**Claim 6:**

Jensen discloses *the method according to claim 3 further comprising the step of employing a standard one of the provisioning API set by the service* (e.g., [0026]-[0030], [0040]-[0042]).

**Claim 7:**

Jensen discloses *the method according to claim 6 further comprising the step of obtaining remotely a custom API via a network coupled to the terminal* (e.g., [0031]-[0034], [0043]-[0046]).

**Claim 8:**

Jensen discloses *the method according to claim 2, wherein the provisioning instructions are selected from the group comprising code, script, and configuration data (e.g., [0027]-[0033], [0037]-[0042]).*

**Claim 10:**

Jensen discloses *the method according to claim 8, wherein the provisioning instructions are separate from the content (e.g., [0014]-[0015], [0024]-[0030]).*

**Claim 11:**

Jensen discloses *the method according to claim 10 further comprising the step of accessing the provisioning instructions remotely from the terminal (e.g., [0014]-[0015], [0043]-[0047], [0050]-[0053]).*

**Claim 12:**

Jensen discloses *the method according to claim 11, wherein the remote access of the provisioning instructions is in conjunction with a networked repository (e.g., [0025], [0033]-[0035]).*

**Claim 13:**

Jensen discloses *the method according to claim 12, wherein the terminal is selected from the group comprising wired devices and wireless devices (e.g., [0027]-[0033], [0037]-[0042]).*

**Claim 14:**

Jensen discloses *the method according to claim 5, wherein a generic API is included in the provisioning API set, the generic API configured for addressing by at least two dissimilar ones of the specified content type (e.g., [0026]-[0030], [0040]-[0042]).*

**Claim 15:**

Jensen discloses *the method according to claim 14 further comprising the step of employing a series of enablers for providing access to corresponding selected ones of the generic API, each of the enablers associated with a predefined content type (e.g., [0024]-[0027], [0040]-[0042], [0047]-[0050]).*

**Claim 16:**

Jensen discloses *the method according to claim 2, wherein a generic API is included in the provisioning API set, the generic API configured for addressing by at least two dissimilar ones of the specified content type (e.g., [0031]-[0034], [0043]-[0046]).*

**Claim 17:**

Jensen discloses *the method according to claim 16 further comprising the step of employing a series of enablers for providing access to corresponding selected ones of the generic API, each of the enablers associated with a predefined content type (e.g., [0027]-[0033], [0041]-[0043]).*

**Claim 18:**

Jensen discloses *the method according to claim 17, wherein the enabler is an executable unit that executes provisioning instruction requests for the predefined content type (e.g., [0028]-[0031], [0036]-[0039], [0041]-[0043]).*

**Claim 19:**

Jensen discloses *the method according to claim 18 further comprising the step of obtaining the enabler selected from the group comprising: locally on the terminal by a provisioning service (e.g., [0025], [0033]-[0035]);*

*bundled with a content descriptor of the content; and remotely from the terminal by the provisioning service (e.g., [0024]-[0027], [0031]-[0033]).*

**Claim 20:**

Jensen discloses *the method according to claim 5, wherein the provisioning instructions were amended prior to the step of obtaining the provisioning instructions by the runtime environment* (e.g., [0014]-[0015], [0024]-[0030]).

**Claim 21:**

Jensen discloses *a terminal, including a computer processor and a computer readable storage medium for providing customized provisioning of an application on a runtime environment* (e.g., FIG. 2, provisioning applications/services from Provisioning Application 208 and Database 220, and deploying/installing provisioned applications/services on target devices 202a-c, [0014]-[0015], [0024]-[0030]),

*the application including content* (e.g., Provisioning Application 208 and Database 220 includes a plurality of contents, [0025], [0033]-[0035])

*having at least one specified content type* (e.g., target devices/user profiles, price/service plans, [0027]-[0033], [0037]-[0042]), *the terminal comprising:*

*a processing framework for obtaining the content* (e.g., FIG. 2, runtime environment of Provisioning Server 204, [0024]-[0027], [0031]-[0033]);

*obtaining by the runtime environment a set of provisioning instructions related to the content type* (e.g., [0027][0033], for each profile, price/service plan → executing/obtaining a set or provisioning instructions associated with said profile, price/service plan, [0037]-[0042]),

*the provisioning instructions being customized* (e.g., [0032] and [0035], for particular client device data and identified price/service plan → specific provisioning instructions are executed, i.e., customized)

*by distributed provisioning control through the provisioning instructions* (e.g., FIG. 2, execution/control of provisioning instructions/APIs have been distributed over Provisioning Application 208, Provisioning API 222, Adapter 206, but not been hardcoded in target devices 202),

*a provisioning API set included in the processing framework for provisioning the content* (e.g., FIG. 3, Provisioning Server 204 specifying either

Discovery, Subscription, or Delivery Provisioning API set, [0028]-[0031], [0036]-[0039], [0041]-[0043]); *and*

*a set of provisioning instructions related to the content, the provisioning instructions coupled to the application for specifying selected ones of the provisioning API set (e.g., FIG. 2, provisioning applications/services according to target devices/user profiles/price plans, [0014]-[0015], [0043]-[0047], [0050]-[0053]).*

Jensen does not explicitly disclose *the provisioning instructions being customized for different subsets of versions of the application.*

However, in an analogous art, Kjellbert further discloses *the provisioning instructions being customized for different subsets of versions of the application (e.g., [0024]-[0026]).*

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine Kjellberg's teaching into Jensen's teaching. One would have been motivated to do so to provision a suitable/new version to client devices in real-time as suggested by Kjellberg (e.g., [0025]-[0026]).

Neither Jensen nor Kjellberg explicitly discloses *executing by the runtime environment the provisioning instructions for employing the API set, by a script interpreter.*

However, in an analogous art, Krantz further discloses *[executing by the runtime environment the provisioning instructions for employing the API set,] by a script interpreter* (interpreting XML-format files/XML provisioning files) as follows:

[0047] Network provisioning services are potentially available via a variety of media (e.g., WWAN, dial-up, DSL, etc.). A user, through selection rule type "3" recited above (network provider service preference order), potentially specifies a rule that defers selection of a particular media to provisioning service-supplied rules. In an illustrative



embodiment, the provisioning service 314 specifies such rules in the form of XML files.

[0063] The following three methods are supported by the common RPC API 303 of the rules engine for the provisioning services 214. A CreateNetworkConfiguration method 434 is similar to the above-described SetNetworkAuthData method 432; however, a network configuration is supplied in XML format via an input parameter. An UpdateNetworkConfiguration method 436 is similar to the CreateNetworkConfiguration method 434 except that a pre-existing configuration is assumed to exist for the identified network. A CreateUserAuthData method 438 is similar to the SetUserAuthData method 428 except the user data is supplied in a passed parameter in XML format.

[0089] In yet another scenario, provided in the fourth row of the table, a rule applied by the rules engine 300 specifies a preference order comprising logical networks. A specific scenario example includes specifying: a corporate network over WISP A over WISP B. A network provider decides which physical network (WLAN over WWAN) to use based upon XML provisioning. Parameters used in such a rule scenario include: XML provisioning files from a wireless provider, business logic to facilitate dynamic network and interface selection. (emphasis added).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine Krantz's teaching into Jensen and Kiellberg's teaching. One would have been motivated to do so to provide network provisioning services by using XML rules files, configuration files, and provisioning files as suggested by Krantz (e.g., [0047], [0063], and [0089]).

**Claim 22:**

Jensen discloses *the terminal according to claim 21, wherein provisioning control of the content is shared between the framework and the application through the coupled provisioning instructions* (e.g., [0027]-[0033], [0037]-[0042]).

**Claim 23:**

Jensen discloses *the terminal according to claim 22 further comprising a provisioning service to direct the provisioning API, the service configured for recognizing the provisioning instructions* (e.g., [0028]-[0031], [0036]-[0039], [0041]-[0043]).

**Claim 24:**

Jensen discloses *the terminal according to claim 23 wherein the service is configured for customizing the provisioning process and the associated provisioning API set according to the provisioning instructions* (e.g., [0014]-[0015], [0024]-[0030]).

**Claim 25:**

Jensen discloses *the terminal according to claim 24, wherein the service is shared by a plurality of the applications* (e.g., [0014]-[0015], [0043]-[0047], [0050]-[0053]).

**Claim 26:**

Jensen discloses *the terminal according to claim 23, wherein the service employs a standard one of the provisioning API set* (e.g., [0025], [0033]-[0035]);

**Claim 27:**

Jensen discloses *the terminal according to claim 26, a custom API is obtained remotely by the service via a network coupled to the terminal* (e.g., [0024]-[0027], [0031]-[0033]).

**Claim 28:**

Jensen discloses *the terminal according to claim 22, wherein the provisioning instructions are selected from the group comprising code, script, and configuration data (e.g., [0026]-[0030], [0040]-[0042]).*

**Claim 30:**

Jensen discloses *the terminal according to claim 28, wherein the provisioning instructions are separate from the content (e.g., [0024]-[0027], [0040]-[0042], [0047]-[0050]).*

**Claim 31:**

Jensen discloses *the terminal according to claim 30, wherein the provisioning instructions are configured for obtaining the remotely from the terminal (e.g., [0031]-[0034], [0043]-[0046]).*

**Claim 32:**

Jensen discloses *the terminal according to claim 31, wherein the remote access of the provisioning instructions is in conjunction with a networked repository (e.g., [0014]-[0015], [0033]-[0035]).*

**Claim 33:**

Jensen discloses *the terminal according to claim 32, wherein the terminal is selected from the group comprising wired devices and wireless devices (e.g., [0027]-[0033], [0041]-[0043]).*

**Claim 34:**

Jensen discloses *the terminal according to claim 25, wherein a generic API is included in the provisioning API set, the generic API configured for addressing by at least two dissimilar ones of the specified content type (e.g., [0026]-[0030], [0040]-[0043]).*

**Claim 35:**

Jensen discloses *the terminal according to claim 34 further comprising a series of enablers for providing access to corresponding selected ones of the generic API, each of the enablers associated with a predefined content type (e.g., [0014]-[0015], [0043]-[0047], [0050]-[0053]).*

**Claim 36:**

Jensen discloses *the terminal according to claim 22, wherein a generic API is included in the provisioning API set, the generic API configured for addressing by at least two dissimilar ones of the specified content type (e.g., [0014]-[0015], [0024]-[0030]).*

**Claim 37:**

Jensen discloses *the terminal according to claim 36 further comprising a series of enablers for providing access to corresponding selected ones of the generic API, each of the enablers associated with a predefined content type (e.g., [0027]-[0033], [0037]-[0042]).*

**Claim 38:**

Jensen discloses *the terminal according to claim 37, wherein the enabler is an executable unit that executes provisioning instruction requests for the predefined content type (e.g., [0024]-[0027], [0031]-[0033]).*

**Claim 39:**

Jensen discloses *the terminal according to claim 38, wherein the enabler location is selected from the group comprising: locally on the terminal by a provisioning service (e.g., [0028]-[0031], [0036]-[0039], [0041]-[0043]);*

*bundled with a content descriptor of the content; and remotely from the terminal by the provisioning service (e.g., [0025], [0033]-[0035]).*

**Claim 40:**

Jensen discloses *the terminal according to claim 25, wherein the provisioning instructions were amended prior to the step of obtaining the provisioning instructions by the runtime environment* (e.g., [0026]-[0030], [0040]-[0042]).

**Claim 41:**

Jensen discloses *a method for providing customized provisioning of an application on a runtime environment of a terminal* (e.g., FIG. 2, provisioning applications/services from Provisioning Application 208 and Database 220, and deploying/installing provisioned applications/services on target devices 202a-c, [0014]-[0015], [0024]-[0030]),

*the application including content* (e.g., Provisioning Application 208 and Database 220 includes a plurality of contents, [0025], [0033]-[0035])

*having at least one specified content type* (e.g., target devices/user profiles, price/service plans, [0027]-[0033], [0037]-[0042]), *the method comprising the steps of:*

*sending the content for receipt by the runtime environment* (e.g., FIG. 2, runtime environment of Provisioning Server 204, [0024]-[0027], [0031]-[0033]);

*obtaining by the runtime environment a set of provisioning instructions related to the content type* (e.g., [0027][0033], for each profile, price/service plan → executing/obtaining a set or provisioning instructions associated with said profile, price/service plan, [0037]-[0042]),

*the provisioning instructions being customized* (e.g., [0032] and [0035], for particular client device data and identified price/service plan → specific provisioning instructions are executed, i.e., customized)

*by distributed provisioning control through the provisioning instructions* (e.g., FIG. 2, execution/control of provisioning instructions/APIs have been distributed over Provisioning Application 208, Provisioning API 222, Adapter 206, but not been hardcoded in target devices 202),

*sending a set of provisioning instructions related to the content for receipt by the runtime environment, the provisioning instructions coupled to the application for specifying a provisioning API set for provisioning the content (e.g., FIG. 3, Provisioning Server 204 specifying either Discovery, Subscription, or Delivery Provisioning API set, [0028]-[0031], [0036]-[0039], [0041]-[0043]); and*

*making available selected ones of the API provisioning set for use by the provisioning instructions; wherein execution of the provisioning instructions employs the API provisioning set to provision the application according to the specified content type (e.g., FIG. 2, provisioning applications/services according to target devices/user profiles/price plans, [0014]-[0015], [0043]-[0047], [0050]-[0053]).*

Jensen does not explicitly disclose *the provisioning instructions being customized for different subsets of versions of the application.*

However, in an analogous art, Kjellbert further discloses *the provisioning instructions being customized for different subsets of versions of the application (e.g., [0024]-[0026]).*

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine Kjellberg's teaching into Jensen's teaching. One would have been motivated to do so to provision a suitable/new version to client devices in real-time as suggested by Kjellberg (e.g., [0025]-[0026]).

Neither Jensen nor Kjellberg explicitly discloses *executing by the runtime environment the provisioning instructions for employing the API set, by a script interpreter.*

However, in an analogous art, Krantz further discloses *[executing by the runtime environment the provisioning instructions for employing the API set,] by a script interpreter* (interpreting XML-format files/XML provisioning files) as follows:

[0047] Network provisioning services are potentially available via a variety of media (e.g., WWAN, dial-up, DSL, etc.). A user, through selection rule type "3" recited above

(network provider service preference order), potentially specifies a rule that defers selection of a particular media to provisioning service-supplied rules. In an illustrative embodiment, the provisioning service 314 specifies such rules in the form of XML files.

[0063] The following three methods are supported by the common RPC API 303 of the rules engine for the provisioning services 214. A CreateNetworkConfiguration method 434 is similar to the above-described SetNetworkAuthData method 432; however, a network configuration is supplied in XML format via an input parameter. An UpdateNetworkConfiguration method 436 is similar to the CreateNetworkConfiguration method 434 except that a pre-existing configuration is assumed to exist for the identified network. A CreateUserAuthData method 438 is similar to the SetUserAuthData method 428 except the user data is supplied in a passed parameter in XML format.

[0089] In yet another scenario, provided in the fourth row of the table, a rule applied by the rules engine 300 specifies a preference order comprising logical networks. A specific scenario example includes specifying: a corporate network over WISP A over WISP B. A network provider decides which physical network (WLAN over WWAN) to use based upon XML provisioning. Parameters used in such a rule scenario include: XML provisioning files from a wireless provider, business logic to facilitate dynamic network and interface selection. (emphasis added).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine Krantz's teaching into Jensen and Kiellberg's

teaching. One would have been motivated to do so to provide network provisioning services by using XML rules files, configuration files, and provisioning files as suggested by Krantz (e.g., [0047], [0063], and [0089]).

**Claim 42:**

Jensen discloses *a computer program product for providing customized provisioning of an application on a runtime environment of a terminal* (e.g., FIG. 2, provisioning applications/services from Provisioning Application 208 and Database 220, and deploying/installing provisioned applications/services on target devices 202a-c, [0014]-[0015], [0024]-[0030]),

*the application including content* (e.g., Provisioning Application 208 and Database 220 includes a plurality of contents, [0025], [0033]-[0035])

*having at least one specified content type* (e.g., target devices/user profiles, price/service plans, [0027]-[0033], [0037]-[0042]), *the computer program product comprising:*

*a computer readable medium; a processing framework module stored on the computer readable medium for obtaining the content* (e.g., FIG. 2, runtime environment of Provisioning Server 204, [0024]-[0027], [0031]-[0033]);

*a provisioning service module coupled to the framework module, the provisioning service module configured for utilizing a provisioning API set for provisioning the content* (e.g., FIG. 3, Provisioning Server 204 specifying either Discovery, Subscription, or Delivery Provisioning API set, [0028]-[0031], [0036]-[0039], [0041]-[0043]);

*obtaining by the runtime environment a set of provisioning instructions related to the content type* (e.g., [0027][0033], for each profile, price/service plan → executing/obtaining a set or provisioning instructions associated with said profile, price/service plan, [0037]-[0042]),

*the provisioning instructions being customized* (e.g., [0032] and [0035], for particular client device data and identified price/service plan → specific provisioning instructions are executed, i.e., customized)



*by distributed provisioning control through the provisioning instructions (e.g., FIG. 2, execution/control of provisioning instructions/APIs have been distributed over Provisioning Application 208, Provisioning API 222, Adapter 206, but not been hardcoded in target devices 202), and*

*an interpreter module coupled to the framework module, the interpreter module configured for executing a set of provisioning instructions related to the content (e.g., FIG. 2, provisioning applications/services according to target devices/user profiles/price plans, [0014]-[0015], [0043]-[0047], [0050]-[0053]),*

*the provisioning instructions associated with the application for specifying selected ones of the provisioning API set (e.g., FIG. 3, Provisioning Server 204 specifying either Discovery, Subscription, or Delivery Provisioning API set, [0028]-[0031], [0036]-[0039], [0041]-[0043]).*

Jensen does not explicitly disclose *the provisioning instructions being customized for different subsets of versions of the application.*

However, in an analogous art, Kjellbert further discloses *the provisioning instructions being customized for different subsets of versions of the application (e.g., [0024]-[0026]).*

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine Kjellberg's teaching into Jensen's teaching. One would have been motivated to do so to provision a suitable/new version to client devices in real-time as suggested by Kjellberg (e.g., [0025]-[0026]).

Neither Jensen nor Kjellberg explicitly discloses *executing by the runtime environment the provisioning instructions for employing the API set, by a script interpreter.*

However, in an analogous art, Krantz further discloses *executing by the runtime environment the provisioning instructions for employing the API set, by a script interpreter (e.g., [0089], XML parser/interpreter as a script interpreter).*

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine Krantz's teaching into Jensen and Kiellberg's teaching. One would have been motivated to do so to provide network provisioning services by using XML rules files, configuration files, and provisioning files as suggested by Krantz (e.g., [0047], [0063], and [0089]).

### **Conclusion**

10. Any inquiry concerning this communication should be directed to examiner Thuy (Twee) Dao, whose telephone/fax numbers are (571) 272 8570 and (571) 273 8570, respectively. The examiner can normally be reached on every Tuesday, Thursday, and Friday from 6:00AM to 6:00PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam, can be reached at (571) 272 3695.

Any inquiry of a general nature of relating to the status of this application or proceeding should be directed to the TC 2100 Group receptionist whose telephone number is (571) 272 2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Thuy Dao/ (Twee)  
Examiner, Art Unit 2192